

OAK LEAVES

ATOM USERS' GROUP CANADA Newsletter #4

The Acorn Atom may no longer be in production, but the drop in price of the basic 2+8k machine to less than \$100 seems to have stimulated users to think expansion. There are various boards available, memory expansion, RS232-C (from PLB), and disk controller for example. Typically these boards cost from \$20 for a bare RS232-C board, through \$35 for a memory expansion or floppy disk controller card (bare, with parts list and directions), up to \$170 for a working RS232-C complete with software or \$200 for an FDC card assembled and tested complete with blank EPROM. If you want more information on these items, which are available from different members, send a stamped addressed envelope to me and I will pass it on to the appropriate member. Please note that these are not available from the Users' Group, but are being offered by individual members.

I would like to thank all those who have sent in material for the Newsletter. It is all greatly appreciated. If you have an item for publication it can be forwarded in many different ways. On disk or cassette, as crisp black copy ready for printing or in any readable form, or posted as a text file on the Burlington bulletin board Tel.(416) 632-5653. If you send it on disk or cassette please do not include special printer control codes. Disks and cassettes will, of course be returned. If you post it on the BBS it is not necessary to format the text. Enter it just as you would in Word-Pack typing <return> only at the end of a paragraph. To signify the end of the text type "/x". The BBS is menu driven. To send a message reply to the prompts with single letter replies as follows:

<E> Enter Mail; <C> Create Message; <S> Save Message; at this point you will be asked for a 3 character code, reply <JCW>. To sign off type <M> Previous Menu; <T> Terminate.

John Wood
February 1984

This Newsletter is published by Atom Users' Group Canada,
812 Cabot Trail, Milton, ONT. L9T 3M8 for the benefit of members
of the Group.

Communications A Zero Cost Serial Interface

One of the most exciting fields in microcomputing today is that of communicating with the outside world. Exchanging data with other computers and accessing the many bulletin board systems that have sprung up really widens your horizons.

In an earlier Newsletter we carried a schematic and program listing for a full RS232-C interface for the Atom. Elsewhere in this issue you will find corrections to this. The corrections were posted to John Wood on a BBS in Toronto by Drew Nisbet the author of the original article in InfoAge. The message was then downloaded to an Atom in Milton, massaged through Word-Pack and printed out as you see it.

The programs listed below enable the Atom to generate serial data suitable for transmission via a modem. No RS232-C board is required. The program emulates the RS232 using the VIA chip and the central processor. Transmitted data is found on PB0, that is pin 10 on the first row of holes on PL6 (start counting from the side of the Atom). Received data is input to PB1 (pin 9 on PL6). The ground for these lines can be taken from pin 32 also on PL6. The signals are in TTL logic, that is "1" is +5V and "0" is 0V. Some but not all modems will accept these signals directly. The system is working very well on an acoustic coupler purchased for \$28.00. Some of these can still be found lurking about in dusty corners around Queen Street in Toronto. (Try Active Surplus). Surplustronics also have some but these have been removed from their great clumsy cases and put in neat little boxes, so they cost twice as much. Note that although these operate on TTL logic they do require +12V and -12V for operation. Also they are "originate only" mode, which means that they are fine for sending and receiving data to and from bulletin boards, or exchanging data with another computer equipped with an "originate/answer" modem. But, they cannot communicate directly with each other.

No apology is made for the untidy structure of the programs. The only justification is that it is guaranteed to be a working version. The main program "comm" requires a lookup table. This is provided by first running LOOKUP. The versions given require a 12 +12k Atom with VIA chip, but they are written in such a way that it is fairly simple to patch them to make use of memory expansions. Again, as written only text files up to 256 (100 hex) characters long can be generated in Word-Pack for subsequent transmission. However, much longer files can be used if "comm" is first run to produce the machine code, the text file is then loaded and "comm" is entered by LINK #3A00. If you do this you will have to make a separate program of lines 1350-1450 to reprint text downloaded to memory. If you wish to create longer files with Word-Pack, you will find that you have to save it as "FILNAM", <BREAK> to get out of W-P then *LOAD "FILNAM". This is to avoid the glitch which occurs at #2900 and #2901 when you push <BREAK>.

There is a trade off in using the 6502 to generate the serial code. It does not leave the CPU much time to do anything else. You cannot send and receive a character at the same instant, although you will pick up the echo when you type. There is not enough time for the program to react to control codes as they are received but they will be stored in memory along with the text. You will find that the screen does not scroll in the normal way. The text always prints down the screen over-writing the previous page as it does so.

```

10 REM COMM
20 REM NEEDS LOOKUP TABLE AT #8200
30 DIMB(4),LL(22),NN(10),JJ(10)
40 F=#3A00
50 F.I=0 TO 22;LL(I)=P;N.I
60 F.I=0 TO 10;NN(I)=P;JJ(I)=P;N.I
70 F.#12$30$21
80 F.I=1 TO 2;F=#3A00
90C
100:JJ1 NOP
110:JJ2 LDA @0
120 STA #B80C;STA #B800;STA #A6
130 STA #80;STA #81;STA #82;STA #A8
140 LDA @1;STA #B802
150 LDA @63;STA #A0
160 LDA @#80;STA #A7
170 LDA @#83;STA #83
180 LDA @#28;STA #A9
190:JJ3 JSR LL1
200:JJ4 RTS
210:JJ5 LDY @0;LDA (#A8),Y
220 CMP @4;BEQ JJ4
230 STA #90;JSR #FE52;JSR LL11
240 INC #A8;BNE JJ6
250 INC #A9
260:JJ6 JMP JJ5
270:LL1 LDA @#3F;JSR #FE52
280:LL2 LDA #B800
290 AND @2;BEQ LL4\NO REC. DATA
300 JMP LL14
310\
320\ CHECK FOR KEYED DATA AND SEND
330:LL4 JSR #FE71 SCAN KEYBOARD
340 CPY @255;BEQ LL2 NO KEYPRESS
350 CPY @59;BNE LL5 IF ESC KEY (BRK)
360 JSR LL20;JSR LL19;RTS
370:LL5 LDA #B001 CHECK FOR SHIFT
380 AND @128;BNE LL6
390 TYA;CLC;ADC @60 ADD OFFSET
400 JMP LL9
410:LL6 LDA #B001 CHECK FOR CONTROL
420 AND @64;BNE LL10
430 CPY @4\IF CTRL-^ (UPLOAD FROM WP)
440 BNE LL3
450 JSR LL20;JSR LL19;JSR JJ5
460 LDA @0;STA #A8
470 LDA @#28;STA #A9;JMP LL13
480:LL3 CPY @1\IF CTRL-C (DOWNLOAD)
490 BNE LL7
500 LDA @1;STA #80\SET DOWNLOAD FLAG
510 STA #81\SET MEMORY FLAG
520 JSR LL20;JSR LL16\PRINT "OK"
530 JSR LL19;JMP LL13
540:LL7 CPY @3\CHECK IF CTRL-J
550 BNE LL8
560 LDA @0;STA #80\TURN OFF DOWNLOAD
570 JSR LL20;JSR LL16\PRINT "OK"
580 JSR LL19;JMP LL13
590:LL8 TYA;CLC;ADC @120 ADD OFFSET
600:LL9 TAY
610:LL10 LDA #8200,Y\GET ASCII
620 STA #90;JSR #FE52
630 JSR LL19;JSR LL11;JMP LL4
640:NN1\DELAY ROUTINE
650 STX #A1;STY #A4;LDY @10
660:NN2 LDX #A0
670:NN3 DEX;BNE NN3
680:NN4 DEY;BNE NN2
690 LDX #A1;LDY #A4
700 RTS
710:LL11\PRINT ROUTINE
720 LDA @1;STA #B800;JSR NN1\START BIT
730 LDA @1;STA #A2;LDX @7
740\INITIALISE PARITY AND BIT COUNT
750 LDA #90;STA #A3
760:NN10 AND @1;EOR #A2;STA #A2
770 LDA #A3;AND @1;EOR @1
780 STA #B800;JSR NN1
790 ROR #A3;LDA #A3
800 DEX;BNE NN10
810 LDA #A2;STA #B800
820 JSR NN1\PARITY BIT
830 LDA @0;STA #B800
840 JSR NN1\STOP BIT
850 JSR NN1;JSR NN1;JSR NN1
860:LL12 RTS
870:LL13
880 LDA @0;LDY @0;JMP LL4
890\
900:LL14 STX #A1;STY #A4;LDY @5
910 JSR NN2\ HALF DELAY
920 LDA #B800;AND @2;BNE LL17
930 JMP LL4\NOT A START BIT
940:LL17 LDA @0;STA #A5;LDX @8

```

```

950:LL15 JSR NN1
960 LDA #B800;AND @2;EOR @2\SELECT BIT
970 LSR A;LSR A\STORE IN CARRY FLAG
980 ROR #A5\ADD BIT TO BYTE
990 DEX;BNE LL15
1000 LDY @6;JSR NN2
1010:NN9 LDA #A5;AND @#7F;STA #A5
1020\IGNORE PARITY BIT
1030 SEC;SBC @#20;BMI NN5\CONTROL CODES
1040 SEC;SBC @#20;BMI NN6\NOS. & PUNCT.
1050 SEC;SBC @#20;BMI NN7\@-^
1060 LDA #A5;CLC;ADC @#20;JMP NN8
1070:NN5 LDA @#20;JMP NN8
1080:NN6 LDA #A5;JMP NN8
1090:NN7 LDA #A5;SEC;SBC @#40;JMP NN8
1100:NN8 LDY @0;STA(#A6),Y
1110 INC #A6;BNE LL21\INCREMENT POINTER
1120 LDA #A7;EOR @1;STA #A7
1130:LL21 LDY #80;CPY @0
1140 BEQ LL18\NOT DOWNLOADING
1150 LDY @0;LDA #A5;STA (#82),Y
1160 LDA #82;CLC;ADC @1;STA #82
1170 BNE LL18\NOT CARRY
1180 LDA #83;CLC;ADC @1;STA #83
1190 CMP @#98;BNE LL18\FREE MEMORY
1200 LDA @0;STA #80\STOP DOWNLOAD
1210:LL18 LDA #B800;AND @2;BNE LL18
1220 JMP LL4
1230\
1240\ SUBROUTINE TO PRINT "OK"
1250:LL16 LDA @#4F ASCII "O"
1260 JSR #FE52
1270 LDA @#4B;JSR #FE52 ASCII "K"
1280:LL20 LDA @#A;JSR #FE52
1290 LDA @#D;JSR #FE52;RTS\CR
1300:LL19 JSR #FE71;CPY @255;BNE LL19
1310 RTS\KEY RELEASED
1320]
1330 N.I;P.#6;LI. JJ1
1340 IF ?#81<>1;END
1350 @=0;P.#12"DATA STORED FROM "
1360 P."#8300 TO #"&(?#82+256*?#83)"/
1370 INPUT"PRINT OUT"#B
1380 IF $B="N" OR $B="NO";END
1390aINPUT"HARD COPY"#B
1400 IF$B="Y"OR $B="YES";P.#2#21
1410 FOR I=#8300 TO (?#82+256*?#83)-1
1420 P.$?I;N.I;P.'/$3#6
1430 INPUT"PRINT AGAIN"#B
1440 IF $B="Y"OR$B="YES";GOTOa
1450 @=8;END

```

"comm" Program Notes

The following locations are used in the program.

?#80 Download flag
 ?#81 Memory use flag
 ?#82-#83 Pointer for download
 ?#90 Storage for byte to send
 ?#A0 Delay to set baud rate
 ?#A1 Temp. storage for X
 ?#A2 Storage for parity bit
 ?#A3 Temp. storage for byte
 ?#A4 Temp. storage for Y
 ?#A5 Received byte
 ?#A6-#A7 Screen pointer
 ?#A8-#A9 Upload pointer

All of these except the pointers could be moved off zero page if required.

If the look-up table is relocated, change line 610 for the new base address.

```

10 REM NON SHIFTED ASCII TABLE
20 !#8200=#5D5C5B20;!#8204=#0B09005E
30 !#8208=#00000000;!#820C=#00000D00
40 !#8210=#33323130;!#8214=#37363534
50 !#8218=#3B3A3938;!#821C=#2F2E2D2C
60 !#8220=#43424140;!#8224=#47464544
70 !#8228=#4B4A4948;!#822C=#4F4E4D4C
80 !#8230=#53525150;!#8234=#57565554
90 !#8238=#1B5A5958
100 REM
110 REM SHIFTED ASCII TABLE
120 !#823C=#7D7C7B20;!#8240=#0A08007E
130 !#8244=#00000000;!#8248=#5F040D00
140 !#824C=#23222130;!#8250=#27262524
150 !#8254=#2B2A2928;!#8258=#3F3E3D3C
160 !#825C=#63626160;!#8260=#67666564
170 !#8264=#6B6A6968;!#8268=#6F6E6D6C
180 !#826C=#73727170;!#8270=#77767574
190 !#8274=#1B7A7978
200 REM
210 REM CNTL ASCII TABLE
220 !#8278=#00000000;!#827C=#00000000
230 !#8280=#00000000;!#8284=#00000000
240 !#8288=#00000000;!#828C=#00000000
250 !#8290=#00000000;!#8294=#00000000
260 !#8298=#03020100;!#829C=#07060504
270 !#82A0=#0B0A0908;!#82A4=#0F0E0D0C
280 !#82A8=#13121110;!#82AC=#17161514
290 !#82B0=#001A1918
300 END

```

```

Screen dump routines are a PAIN!
The problem with trying to get
graphics 4 onto a printer is tha
t the bytes which form the scree
n memory are stored horizontally
as you look at it; that is, The
first byte of screen memory, #8
000, represents the top left eig
ht dots, Y=191, X=0 to 7. But si
nce the print head on most dot m
atrix printers is a vertical ser
ies of pins, bytes sent to the p
rinter from the screen must repr
esent a vertical column of dots,
which, for the first byte of in
formation, would represent bit 7
of the first byte on the top ei
ght lines. Sounds complicated, b
p01>*RUN "SCREEN DUMP"
PLAY TAPE

```

ASSEMBLER SCREEN DUMP ROUTINE:

Screen dump routines are a PAIN! The problem with trying to get graphics 4 onto a printer is that the bytes which form the screen memory are stored horizontally as you look at it; that is, The first byte of screen memory, #8000, represents the top left eight dots, Y=191, X=0 to 7. But since the print head on most dot matrix printers is a vertical series of pins, bytes sent to the printer from the screen must represent a vertical column of dots, which, for the first byte of information, would represent bit 7 of the first byte on the top eight lines. Sounds complicated, but a Basic program to translate horizontal bytes into vertical bytes isn't terribly so. But. A Basic graphics 4 screen dump will take more than five long, dragging minutes. On the other hand this Assembler program to dump graphics 4 to a Gemini 10x printer is somewhat complicated, but has the virtue of taking about 37 seconds to complete its job. This will save you 45 minutes for every 10 screen dumps.

NOTES:

This program is #E2 bytes long and will fit neatly into any page of memory. I like #2800 to #2900 for such things, unless I am using floating point statements, in which case I might use #3B00-#3C00. The program is self contained, so a LINK to #2800 will set ?#FE, turn on the printer, dump the screen at the middle of the page in normal-density mode, reset the printer and turn off the output. The routine will print out the screen AS YOU SEE IT, with no need to reverse it.

After you have entered the Assembler program, by all means save it, but there is no need to tack it on to every program requiring a screen dump. Just *SAVE the code from where the Atom has placed it, eg: *SAVE "DUMP" 2800 28E3. It may then be *LOADED back without your having to reassemble it every time. By changing P in line 130, you can assemble the code in a number of places and have the code ready to *SAVE and *LOAD whenever you want it. The screen printout you see at the top of this article was *RUN into #3B00.

This routine uses 10 bytes in zero page, #80-#89. If the program calling it also uses this space and will loop back after the dump, the addresses must be reset. Alternatively, you might use #90-#99. The toolbox ROM also uses part of this space and might be confused by the values left behind.

#FEFB is the start of the ROM routine to send the byte in the accumulator to the printer. If 2 or 3 are in the accumulator, they won't get to the printer, as the Atom uses these characters to turn the printer output on and off. Whenever I need to send these particular numbers, I use 130 and 131. The eighth (high) bit is unused in printer output, so offsetting by 128 will allow us to send these troublesome digits. Because an accidental value of 3 could turn off the output, I have made sure that the byte sent to the printer at least starts at 128, then increment it as required. Lines 560-630 are used to clean off the row of dots representing #9800-#981F.

Gemini 10x graphics are printed with the high bit at the top of each column. Other printers, such as Centronics, will have the low bit in this position. Conversion should not be difficult. I cannot test the results, but I should think the following changes will do the trick:

*220LDA@1;STA#83

*240-330: These are Gemini control codes which must be changed to match those on your own printer. These set left margin, line spacing and (at VVO) set to graphics normal density mode.

*390LDA@1

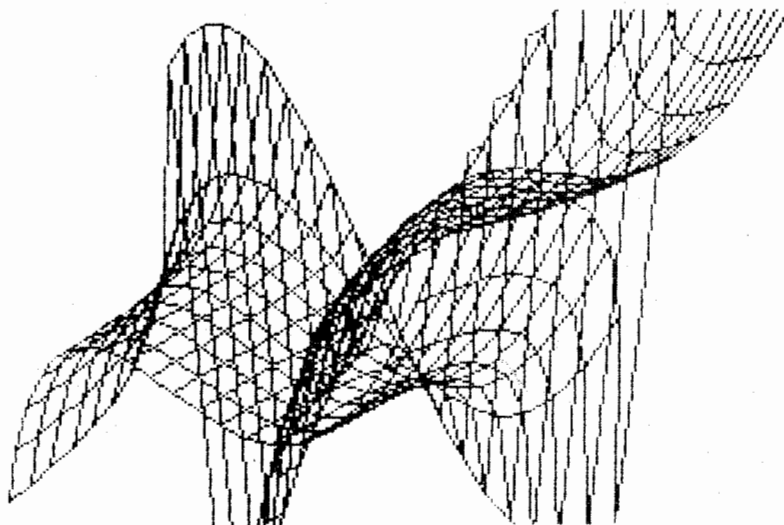
*470:VV4ASL#83

*545CMP@128

*630AND@191

*930-940: Gemini code to reset the printer before exiting the routine. Insert your own printer's reset codes here. Strictly speaking, you may leave these lines out, but subsequent printing will be messed up by the too-short line feed values and too-large left margin.

There may be other patches required. If you work them out, please send them to John Wood.

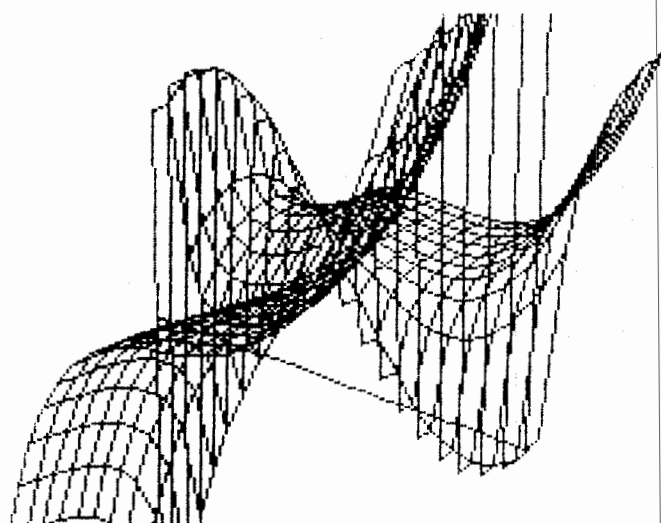


John Lasruk

screen dump
in assembler
by John Lasruk
(for Gemini 10x)

100DIMVV10
110F.\$21
120F.G=0T010;VV(G)=-1;N.
130F.G=1T02;P=#2800
140[
150LDA@3;STA#FE
160LDA@2;JSR#FEFB
170LDA@0;STA#80
180STA#81;STA#85
190LDA@#80;STA#86
200STA#84;STA#89
210LDA@#20;STA#82
220LDA@#40;STA#83
230LDA@#C0;STA#87
240LDA@27;JSR#FEFB
250LDA@77;JSR#FEFB
260LDA@20;JSR#FEFB
270LDA@27;JSR#FEFB
280LDA@65;JSR#FEFB
290LDA@6;JSR#FEFB
300:VVO LDA@27;JSR#FEFB
310LDA@75;JSR#FEFB
320LDA@0;JSR#FEFB
330LDA@1;JSR#FEFB
340LDA#86;STA#81
350LDA#85;STA#80
360LDY@0
370:VV2LDA#89
380STA#88
390LDA@64
400STA#83
410:VV3LDA(#80),Y
420:VV9AND#84
430BNEVV4
440LDA#88
450ORA#83
460STA#88
470:VV4LSR#83
480CLC
490LDA#80
500ADC#82
510STA#80
520BCCVV5
530INC#81
540:VV5LDA#83
550BNEVV3
560LDA#86
570CMP@#97
580BNEVV7
590LDA#85
600CMP@#40
610BNEVV7

620LDA#88
630AND@#FE
640JMPVV10
650:VV7LDA#88
660:VV10JSR#FEFB
670LSR#84
680LDA#86
690STA#81
700LDA#85
710STA#80
720LDA#84
730BNEVV2
740LDA@128
750STA#84
760INY
770CPY@32
780BNEVV2
790LDA@13
800JSR#FEFB
810LDA@10
820JSR#FEFB
830LDA#87
840CLC
850ADC#85
860STA#85
870BCCVV6
880INC#86
890:VV6LDA#86
900CMP@#98
910BEQVV8
920JMPVV0
930:VV8LDA@27;JSR#FEFB
940LDA@64;JSR#FEFB
950LDA@3;JSR#FEFB
960RTS
970J
980N.:F.\$6
990END



SUPERBASIC TOOLKIT

The Superbasic Toolkit is available from MicroPower, 5 Wensley Road, Leeds, LS7 2LX, England at a price of £14.50. According to the manufacturer, this is the final in the series of toolkits for the Atom. It is said to contain most of the original toolkit commands, (it doesn't), and a bunch of new ones. (It does).

The chip is installed in the auxillary ROM socket and is accessed by LINK #AF00 if you wish the 1200 Baud cassette system or LINK #AF04 for the 300 Baud.

As well as the additional BASIC commands, several neat extras are provided as well, as follows.

(A) The Atom bug of printing an error message after control keys has been eliminated.

(B) Several error directives have been corrected.

(C) When an error occurs in Superbasic, the line number is printed, and the cursor highlights the possible area of the error.

New Basic Commands

LOCOS	300 Baud cassette
HICOS	1200 Baud cassette
LEFT\$	
RIGHT\$	Standard Microsoft string handling routines
MID\$	
INSTR	
READ	
DATA	
RESTORE	
WHILE/ENDWHILE	Inverse of DO/UNTIL
FWHILE	F.P. version
XIF-THEN-ELSE	Structured version of IF-THEN
XFIF	F.P. version of above
ON (GOTO/GOSUB)	
INKEY	Scans keyboard for keypress.
BEEP,X,Y	Sounds tone of freq. X, duration Y.
STOP	Stops program at desired point. Any key restarts
ZERO	Zeros all variables
RESET	Pops loop and SR stacks allowing jump outs
ERR	Returns error #
ERL	Returns line # of error
REPORT	Prints error message
ON ERROR	Programmable error handling routines
FREE	Sets free space pointer for DIMs etc
PRINTTAB(X)	Tabs printer
PRINTTAB(X,Y)	Tabs screen in X and Y coordinates
FPRINTTAB	As above in F.P.
SHIFT'B'	Allows number input in binary format
SHIFT'F'	Allows number input in base 4

New Graphic Functions

These are too involved to be fully explained here, but allow shapes to be defined, inverted, and moved about, in colour or B&W.

Additional Error Messages

48 new error messages are added to cope with the new commands.

Summary

The Superbasic kit seems good value. It provides many "standard" basic commands and some interesting graphics. It duplicates 9 of 21 commands in the original Toolkit. As most of the commands in the original Toolkit were concerned with the entering of programs, or manipulating same, e.g. AUTO, RENUM, TRACE, etc., it could be used with the Superbasic Toolkit via a ROM switch, which would give you a very powerful BASIC, without some of the limitations, (memory map problems, etc.) as imposed by the B.E.C. board. Now if only someone would come up with a sensible F.P. ROM that didn't print .1 as 1.0000000E-1!!!

If anyone would like a copy of the instructions for the Superbasic Toolkit for perusal prior to considering purchase of same, send me \$1.00 for postage, (it's 26 pages!), and I will send it to you. My address is:

C.R. Zinck
Lewis Lake, R.R. #3
Armdale, N.S.
B3L 4J3

Time: 01/30/84 05:39:35
From: Fred Springer (KYD)

The Halton Acorn Users' Group met on Tuesday, January 24, 1984 at St. Mark's School on Upper Middle Road in Burlington. There were a few new programs available written by some of the members. In particular, we were able to see a copy program which would copy files from one diskette to another.

The 'hardware' fellows also showed off some of their new goodies which included a RS232 board. As well there was some discussion of making available a board or kit so that a disk could be made up.

Also available that night was a Disassembler/Assembler/Debug chip designed by Alan Hepburn. We didn't get a chance to see it as time was getting late but those who have seen it say it is a neat little package well worth the price that Alan is asking for it. You may contact him if you are interested.

The next meetings will be held on Feb. 21 and Mar. 27, 1984. Meetings begin at 7:30. We hope that you can attend.

DEBUG

An All Canadian Utility ROM

DEBUG is a new utility ROM which does for the assembler programmer what Tool-Box does for the worker in BASIC. Supplied as a 4 k EPROM (2532) it is inserted in IC-24 socket and entered with the command LINK #A000. It comes with excellent documentation which includes examples in addition to easy to follow instructions. DEBUG is menu driven and fully prompted. When it is entered a menu of six options appears:

- 1) INTERACTIVE DISASSEMBLY/DUMP
- 2) DISASSEMBLE TO PRINTER
- 3) DISASSEMBLE TO SOURCE CODE
- 4) BYTE SEQUENCE SEARCH
- 5) DUMP TO PRINTER
- 6) ASSEMBLER LANGUAGE TRACE

On typing 1 the screen shows two boxes with the prompt ADDRESS: All addresses are assumed to be in hex and there is no need to type "#". When entering the starting address to be disassembled DEBUG will only accept valid hex characters plus "DELETE" or "RETURN". When the four character address has been entered the prompt switches to COMMAND: Pressing "^" (up arrow) will cause one op code to be decoded to assembler mnemonics in the left box on the screen. At this stage this box has a bright border around it. Disassembly can be continued with repeated pressing of "^". If shift"^" is pressed ten instructions will be decoded. Replying ">" (no shift needed) will put the bright border around the right hand box. Pressing "^" now causes the contents of the next location to be shown in hex in the second box together with the ASCII equivalent if applicable. Again pressing shift "^" dumps the contents of ten locations. When in dump mode pressing "-" causes a back-step to show the contents of the previous location. Pressing "return" at any stage takes you to the previous prompt. For example if it is pressed whilst stepping through a dump then DEBUG prompts for a new address. Pressing "RETURN" again will step back to the main menu and a third "RETURN" causes an exit back to BASIC. When using option 1 typing ">" or "<" will switch from disassembly to hex/ASCII dump and vice versa.

Option 2 allows a specified block of code to be disassembled and printed out.

Option 3 allows code to be disassembled and stored in a specified block of memory where it can be read and run under BASIC. This allows relocation of code by changing the variable P. References within the range decoded are assigned the labels BB0 to BBn. It is necessary only to add the appropriate DIM statement then run the program twice to generate the relocated code.

Option 4 allows a specified range of memory to be scanned for a string specified in either hex bytes or ASCII. This time ">" and "<" switch to ASCII and back to hex.

Option 5. provides for a dump of memory locations to a printer. This is printed out as 16 hex bytes per line with the ASCII alongside.

Option 6. is a very nifty trick which allows single stepping through machine code. After setting the entry point you are prompted for the entry conditions and can set each register and the flags.

This is all very neat, however some care has to be taken. DEBUG itself uses some zero page addresses from #A0 up and #21C up. If the trace is run on code which uses these same locations there will be confusion. Similarly if you try to single step through operating system subroutines which are used by DEBUG you will get into trouble. To get round this last difficulty, if TRACE encounters a subroutine located above #BFFF this is executed normally returning control to DEBUG when the subroutine call is complete.

The ATOM is way ahead of other small computers in the built in ability to handle assembler language and to teach fundamental machine operation. With this enhancement you will be able to write more code faster and more accurately. Unless you are a professional programmer you probably find that you spend much more time debugging your machine language programs than they took to write in the first place. DEBUG is a tremendous help.

DEBUG is available from:
A. Hepburn
RR#3
Georgetown, ONT.
L7G 4S6

Because of the present unstable situation in the price of chips Alan is offering it for a cost of \$30 if you supply the blank chip. A 2532 which must be by Texas Instruments is required. (This requirement results from a quirk in the ROM blasting board which Alan built for his Atom).

BBS NUMBERS

If you do get your Atom on line, here are a few telephone numbers to try. Only those in Ontario have been tested.

(416) 632-5653 Burlington	(416) 423-5149 E.T.I., Toronto
(416) 524-1973 HACK, Hamilton	(519) 759-5515 Brantford
(604) 738-2773 CoCo, Vancouver	(403) 287-3638 Shop, Calgary
(204) 942-1109 VE4, Winnipeg	(514) 622-1274 Conn80, Montreal

Many of these boards operate only from evening to morning, and there is no guarantee that they are still in operation.

The following message was downloaded from the ETI BULLETin board in Toronto. It refers to the article "Getting On Line" in NL-#4.

Message 22 is from DREW NISBET, to JOHN WOOD.
It's about RS232 FOR ATOM.

John:

The errors in the schematic were:

- 1) The pins on the 1489 are backwards.
Those shown as inputs are outputs and vice versa.
- 2) Pin 26 on the 6551 should not be connected to the expansion port of the ATOM (line 28).
- 3) The chip select lines for the 6551 are incorrect. Line 27 of the Bus Extension should be directly connected to pin 2 of the 6551. The output of the 74LS260 should be input to the 74LS04 and the output of the 74LS04 should be connected to pin 3 of the 6551.

I hope this information will help. In addition, any lines which are not available for any specific MODEM must be tied either high or low depending on the default value for the signal concerned (not just left unconnected).

Drew.

Note that the above refers to the original article in Info-Age. Item (1) had been corrected in the version given in NL-#4.

WORD PROCESSOR TRICKS

The Atom word processor ROM is a *wondrous* little fellow indeed: apart from the display, he is *almost* a match for those fancy disc based jobs on *other* machines. I say *almost* because, in theory, one can only supply control codes to the printer at the beginning of lines, meaning single-word *italics*, **emphasis** or underlining are out. Now my Gemini 10x is *dying* to do all that stuff (and more), so it goes without saying that being unable to directly address the beast in the middle of a line has proved frustrating to say the **least**.

You have the same frustrations too? You hate the fact that it is **impossible** to introduce printer commands in mid-line? Have a *close* look at the previous paragraph and this one. AHA! Here's the trick:

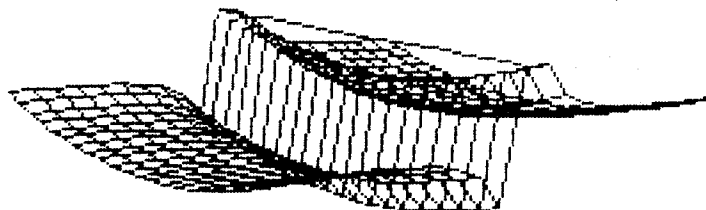
First, enter your text in a straightforward way, with formatting set, but no attempt to enter control codes (the word processor doesn't recognize them in enter mode). Run a hard copy, if you wish, to decide just where underlining, italics, etc. will be needed. Don't get carried away; too many special effects will make a letter look like a dog's breakfast. Insert the proper <ESC> codes before and after the word you have chosen to change, substituting a seldom-used character, such as <~> (shifted up-arrow) for the <ESC> character. Actually, any character not used in the text will do.

Next, do a search and replace, substituting the escape code (27) for the stand-in. For example, if I choose to italicize the word "italics", first I must surround it with the proper select and deselect italics code (in this case for the Gemini; other printers will have other sequences): ~4italics~5. Next, I will enter the word pack command F:, thus: F\~\<ESC>\a. Use CTRL-[to print code 27 (ESC).

Now you will see why I take this somewhat round-about route instead of simply inserting <ESC> where needed. The display becomes very corrupt. It obviously hates characters it cannot print but which exist in letter count. It becomes angry. The cursor gets lost and starts crying. Leave all your playing around until the VERY end. Corrupt as it is, the mean, nasty display cannot affect your wonderful artistic printout.

Now, I hear you asking, "What's the catch? What are the limits on my creativity here? Can you really hear me asking this? What am I doing talking like this?" Okay. Justification is messed up by the countable but unprintable control codes. The first two paragraphs of this article are in theory justified! You can use this technique to enter any control code except CTRL-B or CTRL-C, which the Atom refuses to send to the printer. Also, if you set ?#FE=0, you will be unable to get any reaction to CTRL-@ from the printer, and since the NUL string sometimes is useful, why not set ?#FE=3, since it isn't going anywhere, anyway, anyhow? Secondly, printer commands which require numbers, for example, one which would set a one time line feed of x lines, will need a control character representing the number required, ie: CTRL-D for the number 4, and if you want to send the numbers 2 or 3, forget it. For that matter, forget anything over number 31, because you aren't going to call it up from the keyboard.

So there it is: a handy trick not to be overdone, but adding a useful extra to the very useful word processor ROM. As for me, all this typing has made me thirsty, so I'll just turn off the Atom and have a glass of good old H₂O. Cheers!



John Lasruk

DOS UTILITIES

Several useful DOS utilities have been sent to the Users' Group. Three listed here are originally by Alan Hepburn. Note that as written they require memory expansion above #4000. Note also that they contain self modifying code and some sections must be entered exactly as printed.

FILCOPY

```

10REM COPY A FILE USING A SINGLE DISK DRIVE
11REM CONTAINS SELF MODIFYING CODE
12REM ENTER EXACTLY AS TYPED
15?#D=0;?#E=#2F;?#23=0;?#24=#2F
20DIMS32,N32,Q2;D=0;B=#4000
22P."COPYING A SINGLE FILE""USING THE DEFAULT DRIVE""
30IN."FILE NAME"$S
31P."INSERT SOURCE DISK""AND PRESS SPACE BAR"
32LINK#FE94
34*DIR
40GOS.F;IF F;G.100
50P."NOT FOUND";E.
100F=F+#100
110?#A2=F?6;?#A3=F?7;REM 1ST SECTOR
120?#A0=0;?#A1=(F?5)+1;REM # OF SECTORS
130?#9C=B;?#9D=B/256;REM BUFFER
140LINK#E75B;LINK#E4A3;REM READ THE FILE INTO BUFFER
142$N="*SAVE"*****"XXXX XXXX XXXX"
144L=256*(F?5)+(F?4)
145?Q=?F;Q?1=F?1;REM STARTING POINT
146A=F?3;I=24;GOS.x
148A=F?2;GOS.x
150P."INSERT TARGET DISK""AND PRESS SPACE BAR"
160LINK#FE94
190F.I=0TO6;N?(I+6)=S?I;N.
200A=B/256;I=14;GOS.x
210A=B&#FF;GOS.x
220B=B+L;A=B/256;I=19;GOS.x
230A=B&#FF;GOS.x
2900=46
300J=?5+256*?6+0;F.I=0TO LENN;?J=N?I;J=J+1;N.
310*SAVE"FILCOPY"4000 4000 2000
315*DIR
320GOS.F;F=F+#100;?F=?Q;F?1=Q?1
330LINK#E75B;LINK#E74A;P."OK";E.;REM WRITE INDEX
1000?IF LENS>6;G.1005
1003F.I=LENS TO 6;S?I=#20;N.
1005S?7=#D;F=0
1010F.I=#2008 TO#2000+?#2105 STEP 8
1020F.J=0TO7;N?J=I?J;N.;N?7=#D
1030IF$S=$N;F=I
1040N.;R.
1100C=A/16+#30;GOS.1150
1120C=(A&#F)+#30;GOS.1150
1140R.
1150IFC>#39;C=C+7
1160N?I=C;I=I+1;R.

```

RENAME

```

10REM RENAME DISC FILE
15?#D=0;?#E=#2F;?#23=0;?#24=#2F
20DIMS32,N32
30IN,"OLD NAME"$S
34*DIR
40GOS.f;IF F;G.100
50P,"NOT FOUND";E.
100IN,"NEW NAME"$S
105IF LENS>7;P,"TOO LONG";G.100
110=F;GOS.f;IF(F<>0);P,"IN USE";G.100
120F.I=0 TO LENS -1
1300?I=S?I
140N.;IF I=7;G.160
150F.I=I TO 7;0?I=#20;N.
160LINK#E75B;LINK#E74A;P,"OK";E.
1000?F.I=LENS TO 6;S?I=#20;N.;S?7=#D
1005F=0
1010F.I=#2008 TO#2000+?#2105 STEP 8
1020F.J=0 TO 7;N?J=I?J;N.;N?7=#D
1030IF$S=$N;F=I
1040N.;R.

```

DISCOPY

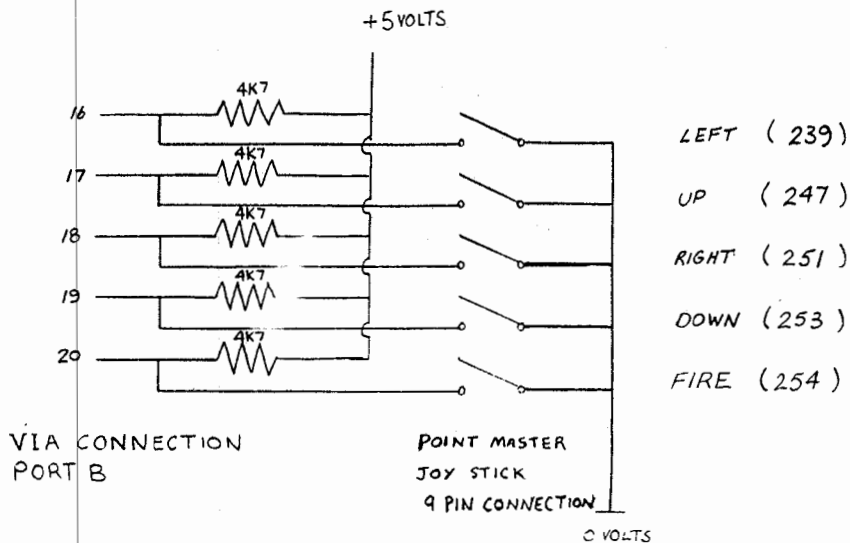
```

1?#D=0;?#E=#2F;?#23=0;?#24=#2F
5DIMS32,B0;D=0
10P.#12"COPY WHOLE DISC"
15IN,"FROM DRIVE 0 TO DRIVE 1, OK"$S
17IF?S=CH"N";P,"OK, FROM DRIVE 1 TO DRIVE 0";D=1
30IN,"READY"$S
40IF?S<>CH"Y"AND S?1<>CH"E"AND S?2<>CH"S"AND S?3<>#D;E.
50P,"copying";@=3
80?#9C=B;?#9D=B/256
90?#A0=2560;?#A1=10
100F.I=0 TO 39
105P,"TRACK" I'
110?#A2=I/256;?#A3=I
120IFD=0;*DR.0
125IFD=1;*DR.1
130LINK#E75B;LINK#E4A3
140IFD=0;*DR.1
145IFD=1;*DR.0
150LINK#E75B;LINK#E6B0
160N.;P,"OK";E.

```


JOYSTICKS

Since the early days of the Atom in Canada, people have plotted ways to add a joystick. Scott McLellan of Ottawa has sent in one of the neatest ideas seen so far. He has also included routines to allow it to be used in your own program and a machine code patch to convert SNAPPER. Scott uses Point Master type joysticks which he bought for \$9.95 U.S. They can be used on Port A and Port B of the VIA but printer output must be disabled to use Port A.



ONLY THE LOWER 5 BITS OF PORT B ON THE VIA CHIP ARE USED
45° MOVEMENT IS POSSIBLE
CANNOT BE USED ON PORT A AT THE SAME TIME AS A PRINTER

5 REM GAME PADDLES CONTROL

```
10X=128;Y=93
200N ERROR G.40
30CLEAR4
40?#B80C=#00
50?#B802=#00
60A=?#B800
70GOS.A;PLOT14,X,Y
80G.60
231Y=Y+1;X=X-1;R.
237X=X-1;Y=Y-1;R.
239X=X-1;R.
243X=X+1;Y=Y+1;R.
247Y=Y+1;R.
249X=X+1;Y=Y-1;R.
251X=X+1;R.
253Y=Y-1;R.
254CLEAR4;R.
255R.
```

MACHINE CODE PATCH TO SNAPPER FOR JOYSTICKS

3800 EA	381D EA	383A EA	3860 F0 0E
3801 EA	381E EA	383B AD 01 B8	3862 A0 00
3802 EA	381F EA	383E 29 02	3864 4C 67 38
3803 EA	3820 AD 01 B8	3840 CD F7 28	3867 8C F3 28
3804 EA	3823 29 04	3843 F0 0C	386A A9 90
3805 AD 01 B8	3825 CD F6 28	3845 8D F7 28	386C 8D 00 B0
3808 29 10	3828 F0 0C	3848 C9 02	386F 60
380A CD F4 28	382A 8D F6 28	384A F0 05	
380D F0 0C	382D C9 04	384C A0 03	
380F 8D F4 28	382F F0 05	384E 4C 67 38	
3812 C9 10	3831 A0 02	3851 AD 01 B8	
3814 F0 05	3833 4C 67 38	3854 29 08	
3816 A0 01	3836 EA	3856 CD F5 28	
3818 4C 67 38	3837 EA	3859 F0 15	
381B EA	3838 EA	385B 8D F5 28	
381C EA	3839 EA	385E C9 08	

QUERY ?

Edouard Dion wants to know if he can use a QumeTrack double sided disk drive with his Atom. The Qume uses 20 ms track to track head access time. Write to him at 3160 Riviere Cachee, Boisbriand, Quebec, J7H 1E2. or pass the information to all of us through the Newsletter.

HELP!!!!!!

If anybody has data on the R6551 ACIA chip please send it to me ASAP. I need it desperately!!!

Colin Hinz 1118 College Dr. Saskatoon, SASK. S7N 0W2

FOR SALE

ATOM 12K + 12K, COLOUR BOARD, 4 POSITION ROM SEL. BOARD, TOOL-KIT & WORD-PACK, VIA & PRINTER INTERFACE. ALL BUFFERS, PWR SUPP, A.T.&P. Package: \$450 or will sell separately for 30% off retail prices. Grant McKenzie 2506 Kenney St. Terrace B.C. V8G 3E3. (604) 635-2547

Acorn Atom - 12K RAM + 12k ROM, Pwr Supply, Toolbox, Printer interface, External bus expansion, Software (including Forth), & Books. \$400. Steve Gibson, 1121 Spears Road, Fort Erie, ONT., L2A 4N2 Phone: (416) 871-3265 -- after 6pm.

ACORN ATOM: 16K RAM, 12K ROM, ROAM BOARD, ATOMCALC, TOOL BOX, MONITOR OUTPUT, COLOUR BOARD, POWER SUPPLY, SOFTWARE: CHESS, ATOMSTORE, LOTS OF GAMES, MISC PROGRAMS, 3 BOOKS. Doug McLaren, Box 66 Smithville ONT. L0R 2A0. (416) 957-7781

ATOM 12+12, B&W, SUPERBASIC, BBC CARD, EXPANSION UNIT C/W EXP. BOARD 16K TIMEDATA MEM., H.D.P.S., T.T.Y. I.O., ALL SCHEMATICS, ALL AVAIL. BOOKS, S/WARE INCLUDES DISASS., GOLF, DATABASES, ELIZA ETC. VAL \$900 SELL FOR \$450. R. ZINCK, LEWIS LAKE, RR3 ARMDALE N.S. B3L 4J3

COLOUR BOARD \$40.00

J.H. Termeulen, RR#4 Trenton, ONT. (613) 392-9864 after 6pm.

Brand New (factory replacement) BBC/Atom Disk Drive. Complete with 3' connecting cables. Great for use as a second drive, or use with an FDC card for a starter drive. \$270.00 O.B.O. Colin Hinz 1118 College Dr. Saskatoon, SASK. S7N 0W2

Hundreds of 2114L memory chips! Guaranteed to work, only \$1.10 each or ten for \$10.00 Also selling, hundreds of other TTL, MOS, and linear chips (not all tested though). Send SASE for free list. Colin Hinz 1118 College Dr. Saskatoon, SASK. S7N 0W2

64k Duo 1 Memory expansion board. Never installed, but sold as is. L.J. Howarth 3-2242 Upper Middle Rd. Burlington, ONT. L7P 2Z9

Non commercial advertisements, which are published free of charge can only be accepted from current members of the Users' Group and will be restricted to a maximum of 4 lines.

NOTICE-

The Halton Acorn Users' Group continues to meet on the third Tuesday in each month at St. Mark's School on Upper Middle Road in Burlington, ONT. BUT! note change for March when it will be the fourth Tuesday due to Winter break.

